

Parallel shooting methods for finding periodic steady state solutions to models of machines with reciprocating pistons

Stig Kildegård Andersen ^{a,*}, Per Grove Thomsen ^b, Henrik Carlsen ^a

^a *Department of Mechanical Engineering, Energy Engineering Section, Technical University of Denmark, Denmark*

^b *Informatics and Mathematical Modelling, Technical University of Denmark, Denmark*

Received 6 April 2005; received in revised form 14 March 2007; accepted 15 June 2007

Available online 22 June 2007

Abstract

Parallel single- and multiple shooting methods were tested for finding periodic steady state solutions to a Stirling engine model. The model was used to illustrate features of the methods and possibilities for optimisations.

Performance was measured using simulation of an experimental data set as test case. A parallel speedup factor of 23 on 33 processors was achieved with multiple shooting. But fast transients at the beginnings of subintervals caused significant overhead for the multiple shooting methods and limited the best speedup to 3.8 relative to the fastest sequential method: Single shooting with reduced dimension of the boundary value problem.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Shooting methods; Parallelisation; Periodic steady state; Stirling engine model

1. Introduction

Computer simulation is an important design tool in the field of gas cycle machines with reciprocating pistons, such as Stirling engines and Stirling coolers and pulse tube coolers (PTCs) [1–3]. In these machines the heating and cooling of the working gas needed in the thermodynamic cycle of the machine is achieved by forcing the gas through heat exchangers and by varying the pressure. The working gas itself does not take part in any combustion or other chemical reactions. Due to the nature of the gas flows in the heat exchangers of the machines it is possible to accurately model many phenomena in the machines using models where the machines are discretised in one space dimension. The one dimensional models use empirical correlations for calculating heat transfer and flow friction and they are constructed as systems of first order ordinary differential equations (ODEs) or as differential algebraic systems of equations (DAEs). Solutions to the models have a periodic nature due to the reciprocating motions of the pistons in the machines.

* Corresponding author. Tel.: +45 6617 6020.

E-mail address: sk.andersen@danfoss.de (S.K. Andersen).

Stirling machines and PTCs are usually designed for optimal operation at steady state conditions and hence periodic steady state solutions are needed when simulation is used for design optimisation of the machines. The problem is that solutions to detailed models that mimic the thermodynamic behaviour of both the gas and the steel in the machines will evolve slowly and asymptotically towards periodic steady state when they are integrated as initial value problems (IVPs). A model may require the simulation of several minutes running time for the engine or cooler in order for a solution to evolve to within the desired tolerance from the true periodic steady state solution. This can mean that a sequence of several thousand revolutions of the machine must be simulated in order to compute a single periodic steady state solution.

The work on the shooting methods described in this paper was initiated to satisfy a need to find periodic steady state solutions to an early one dimensional model of the 9 kW Stirling engine, SM5 [4]. In this early model, which we described in Ref. [5], the inertia of the gas was not included in governing equations for the gas domain. Once the shooting methods had been implemented the model was enhanced to include the inertia of the gas. We described this improved method for modelling the gas in Ref. [6] where it was shown that the new Stirling engine model produces results in good agreement with experimental data.

In the early model, where the inertia of the gas was neglected, any bumps in the initial spatial pressure distribution would equalise and disappear within a few microseconds when the model was integrated as an IVP from a bumpy initial pressure distributions. In the new model the inertia of the gas caused acoustic waves to be initiated when simulations were started from bumpy initial pressure distributions. Tracking the fast transients while these acoustic waves dissipated required a very fine temporal discretisation in the beginning of simulations. The model applies artificial dissipation as described in Ref. [6] in order to accelerate the dissipation of acoustic waves, but depending on the initial magnitude of the acoustic waves they could take several milliseconds to dissipate.

For the new model it was observed that the multiple shooting method became significantly slower and did not always converge very well. There thus appeared to be an unfortunate interaction between the new model and the multiple shooting method used for finding periodic steady state solutions. It was found unacceptable to go back to neglecting the inertia of the gas, because this would make the modelling approach less general and could limit its usefulness for other types of machines such as PTCs. Therefore work was continued with the new model formulation, and therefore the model which includes the inertia of the gas was used as test case for this study.

1.1. Methods for finding periodic steady state solutions

Several methods that can be applied to periodic boundary value problems (BVPs) can be used to find periodic steady state solutions to Stirling machine models. These methods include harmonic analysis, integration to convergence, extrapolation techniques, finite difference methods, and shooting methods. The type of model that is considered in this study consists of a mix of ODEs with periodic solutions and ODEs that are used as integral conditions to determine wall temperatures that are constant during simulations.

When integration to convergence, a.k.a. fixed point iteration, is applied a model is integrated as an IVP for many consecutive cycles of the simulated machine until the solution for the periodic variables has evolved to be within some tolerance of the periodic steady state solution. This method can be considered as the brute force approach, as no special measures are taken to overcome the slow asymptotic approach towards periodic steady state.

The number of cycles that must be simulated for the periodic variables to approach periodic steady state can be reduced by applying general extrapolation techniques such as the ε -algorithm discussed by Skelboe [7,8]. In each iteration of the ε -algorithm a number of consecutive cycles are simulated. From the initial values of each consecutive cycle an extrapolation to the periodic steady state solution is then made by assuming that all deviations from the periodic steady state solution decay exponentially.

In some cases it is also possible to apply convergence acceleration based on known properties of the periodic steady state solution. Kühl [9] demonstrated a method for accelerating just the temperature profile of the regenerator matrix in a Stirling engine towards periodic steady state. The method relies on the fact that if it can be assumed that the regenerator matrix is thermally insulated from its surrounding canister, then at periodic steady state the enthalpy flows through all cross sections of the regenerator matrix will be identical and,

hence, independent of the axial position. Kühl used this fact to derive an iterative method for correcting the regenerator matrix temperatures to make the enthalpy flows through cross sections of the regenerator matrix more similar.

A common advantage of the integration to convergence, extrapolation, and convergence acceleration methods is that their interface to the model is through an IVP method. This means that any events internally to the cycle, such as discontinuous derivatives in the solution to the governing equations caused by moving meshes [10], can be handled by the IVP method so that normally the events need not be considered in the method used for solving the periodic BVP. A common disadvantage of the methods is that they do not solve integral conditions. This means that for the Stirling engine model used in this study the methods would need to be used in a hybrid form using an external solver to solve the integral conditions.

For harmonic analysis the idea is to make sufficient simplifying assumptions in the governing equations of a model to allow the equations to be solved analytically for harmonic solutions. This approach has been explored by NASA during the 1980s and early 1990s [11,12]. The approach leads to fast simulations but is restrictive with regards to how models can be formulated. Also the approach does not handle events internally in the cycle, and integral conditions need to be solved separately.

Finite difference methods are more widely used for Stirling machine models [2,13–16]. The finite difference formulations used by Gedeon [13,14] in the state of the art simulation program *Sage* discretises the governing equations in both time and space to produce a globally implicit finite difference equation system. This equation system is solved directly for a periodic steady state solution expressed in harmonic series. This approach has the advantage that the stability of the numerical scheme can be maintained with as little as 10 temporal grid points in the cycle, which is good when computing approximate solutions. On the other hand the finite difference equation system can become very large and cumbersome if accurate solutions are computed with many spatial and temporal grid points. Convergence problems are common due to the stiff and non-linear nature of the models and good initial guesses are needed to achieve convergence [14]. Convergence and stability of finite difference methods can be especially problematic at points of flow reversal in the solutions [15]. Finite difference methods can be made so that they can handle events internally in the cycle by using one sided differences from both sides towards a temporal mesh point placed exactly at the event. But this is not straightforward for all Stirling machine models because the number of events and their positions inside the cycle may change from iteration to iteration. To our best knowledge the usual approach is to either restrict the models to not having events or to ignore the events in the finite difference method [16].

For shooting methods the cyclic boundary conditions and the integral conditions needed to define the periodic steady state solution is formulated as a system of residual equations. To compute the residuals the model is integrated as an IVP for one cycle of the simulated machine. The system of residual equations, which is typically non-linear, is solved for the initial values of the periodic variables and the parameter values corresponding to the integral conditions. The point of this is that we can change the rate of convergence towards periodic steady state away from the slow asymptotic evolution an IVP into the rate of convergence for a solution process for a non-linear equation system.

Previous work by Commiso [16], who compared integration to convergence, finite difference methods, and the shooting method, indicates that shooting methods can be competitive in terms of CPU-time for finding periodic steady state solutions to models of Stirling machines. It is also attractive properties of shooting methods that events can be handled internally in an IVP method and that the shooting methods cause no restrictions on what can be included in models. Also the shooting methods are uncomplicated to implement and can be parallelised with a coarse granularity suitable for execution on distributed memory platforms. For these reasons the shooting methods were selected for this work.

1.2. Suitability of shooting methods for periodic problems in general

The suitability of the shooting method for any specific periodic problem depends on how difficult and demanding it is to solve the non-linear system of residual equations for the cyclic boundary conditions and integral conditions. If the system of residual equations is highly non-linear or if it has a large number of variables compared to the number of cycles that would be needed for the solution to evolve to the periodic steady state, then using the shooting method might not reduce the CPU-time needed to find periodic steady state

solutions. One can also imagine problems where an unfortunate guess by the shooting method for the initial values and parameters could make it impossible to integrate the IVP. In our experience such issues are generally not problematic for models of Stirling machines and pulse tube coolers except for a few extreme cases.

One problematic case is when the gas domain of the simulated machine is so long that the influences of perturbations of initial values on the residuals for the boundary conditions do not manifest themselves in a single simulated cycle. In this case the cure can be to simulate several consecutive cycles for each evaluation of the residuals. This is usually not relevant for models of practical machines.

Another case is when the equations of motion for flexibly suspended pistons and for a flexibly suspended main body are included in the model and the guess for the initial values is very poor. In this case the pistons and the main body of the simulated machine will oscillate violently as a multi-body system and the pistons can impact on the ends of cylinders. This motion makes the solution so non-linear that the shooting method cannot find solutions just by looking at how residual change in a single cycle. The good thing about the shooting method in such cases is that it is usually possible to determine what is going on by looking at the transient solution from the integration of the IVP. In this case the cure is to let the solution evolve as an IVP until the solution has settled down enough that the shooting method can work.

It is difficult to make a general statement on when shooting methods are competitive for finding periodic steady state solution but as examples they have been used successfully to problems as diverse as Stirling machine models [16], models of electrical circuits [17], and finite element models of electromechanical devices [18].

1.3. *The current study*

This paper presents single and multiple shooting methods for finding periodic steady state solutions to models of machines with reciprocating pistons. The methods are presented using a Stirling engine model as an example to illustrate the features required in the methods. A method that uses the different time scales of the physical phenomena in the models to reduce the dimension of the BVP solved by a single shooting method is also presented. The shooting methods have been parallelised and the computational expense, the convergence, and the scalabilities of the methods on parallel computer platforms are discussed.

2. Background: a Stirling engine model

The equation system that is used throughout this paper for illustration and for testing of the shooting methods is introduced in this section. The equation system is a model of a Stirling engine and in the simulations performed in this paper the model has been configured to model the 9 kW Stirling engine, SM5. A drawing of the gas filled working volume of the SM5 Stirling engine is shown in Fig. 1.

2.1. *The Stirling engine*

Practical Stirling engines basically consist of two cylinder volumes connected by a serial connection of three heat exchangers, i.e. a cooler, a regenerator, and a heater, and the engines have a cold end and a hot end. In the SM5 engine the two cylinder volumes are placed in the same cylinder and are separated by a displacer piston. In the engine cycle work is first expended to compress the gas in the cold end of the engine. The gas is then pushed through the serial connection of heat exchangers to the hot end of the engine. This adds energy to the gas and causes an increase in pressure. The gas is then expanded in the hot end of the engine. Due to the higher pressure this expansion releases more work than was needed for the compression of the gas in the cold end of the engine. To complete the cycle the gas is pushed back through the serial connection of heat exchangers towards the cold end of the engine. This removes energy from the gas and causes a decrease in pressure.

The regenerator, that connects the cooler and heater, is an annular void filled with a porous matrix made from metal felt. It acts as a thermal sponge that absorbs energy when gas is flowing towards the cold end of the engine and releases energy to the gas when the gas is flowing towards the hot end. The primary functions of the regenerator are to limit the amount of energy carried directly from the heater to the cooler by the working gas and to increase the amount of heat transfer and thereby the increase the pressure difference between the compression and the expansion of the working gas.

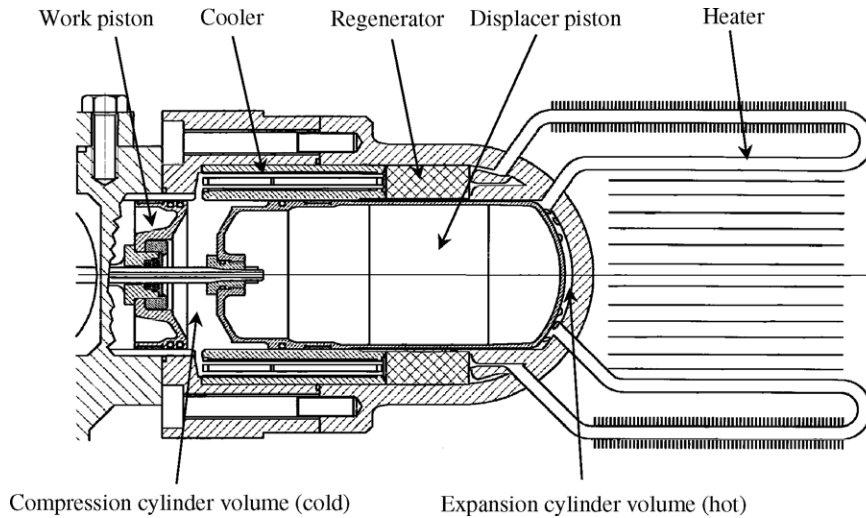


Fig. 1. Working volume of the Stirling engine SM5.

The pressure losses and the heat exchange that takes place in the serial connection of heat exchangers are central to the performance of a Stirling engine. A distributed model of the heat exchangers is needed because temperature fronts travel back and forth through the heat exchangers during the cycle. But due to the nature of the flow in the heat exchangers one dimensional models that use empirical correlations for calculating heat transfer and flow friction are often sufficient to achieve an accurate description of the thermodynamic cycle in a Stirling engine.

2.2. The Stirling engine model

In the one dimensional Stirling engine model used here the internal volume of the engine is represented by the one dimensional discretisation illustrated in Fig. 2.

For the gas in the control volumes in Fig. 2 we use the conservation laws for mass and energy to obtain two first order ordinary differential equations pr. control volume for the time rates of change of the mass and energy contained in the control volumes, $\frac{dm}{dt}$ and $\frac{dE}{dt}$. Using the ideal gas equation of state, thermodynamic relations, and simplifying assumptions about kinetic and potential energies we express the time derivatives of the pressures and temperatures in the control volumes, $\frac{dp}{dt}$ and $\frac{dT}{dt}$, in $\frac{dm}{dt}$ and $\frac{dE}{dt}$. We then apply conservation of momentum to a second set of control volumes that are placed so that their centres coincide with the boundaries of the control volumes in Fig. 2, i.e. to a staggered mesh of control volumes. From the resulting equations for momentum for the staggered mesh of control volumes we then isolate the time derivatives of the velocities at the boundaries of the control volumes in Fig. 2, $\frac{dV}{dt}$.

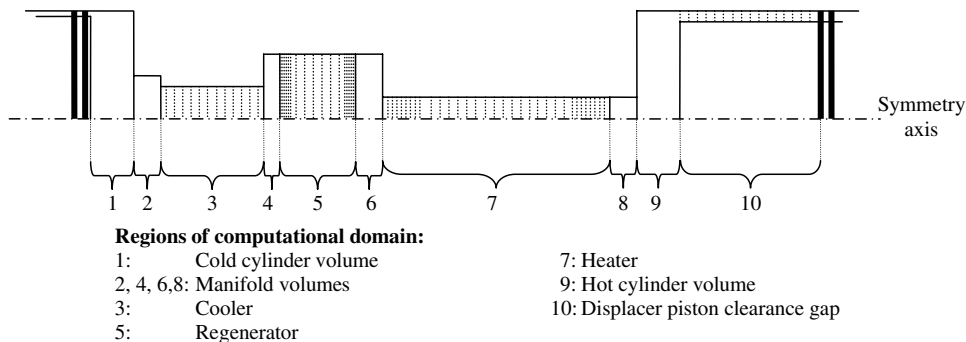


Fig. 2. Discretisation of Stirling engine in one dimensional model.

We also need to describe the temperatures of the steel in the engine. In the regenerator we use conservation of energy for a lumped control mass, i.e. a control mass with a uniform temperature, to obtain expressions for the first order time derivatives of the matrix temperatures in each of the control volumes in the regenerator, $\frac{dT_m}{dt}$. The remaining wall temperatures in the engine, T_w , are assumed fixed in time at their steady state cycle mean values, which are unknown until a solution to the model is computed.

2.3. Conditions for periodic steady state

If an engine is operating at periodic steady state conditions then the values of the periodic variables p , T , V and T_m will remain unchanged from cycle to cycle if observed at the same relative positions in the cycles. This condition can be formulated as periodic boundary conditions for these variables. The net heat transfer to wall segments within the engine will be zero if integrated over a cycle. This requirement can be formulated as integral conditions for T_w . We choose to formulate our model in the periodic variables p , T , V and T_m and in the net amounts of energy transferred to the wall segments with constant temperatures, E_w . We calculate the numerical values in E_w by integrating the time rates of heat transfer to each wall segment. We now have a set of conditions and a system of ODEs that we, given initial values p_0 , T_0 , V_0 , and $T_{m,0}$ and values for T_w , can integrate as an IVP. The system of ODEs has a single infinity of periodic steady state solutions that satisfy the periodic boundary conditions and integral conditions because the requirements for periodic steady state can be fulfilled for any total amount of gas contained within the working volume of the engine. To get a unique solution we add an additional condition for the mean pressure p_{mean} at a point inside the engine. We expect p_{mean} to be more or less proportional to p_0 , but since we also expect the periodic steady state initial values T_0 , V_0 , and $T_{m,0}$ and p_{mean} to be interdependent we need to simultaneously solve for the values of p_0 , T_0 , V_0 , $T_{m,0}$, and T_w that yield the desired value of p_{mean} and satisfy the conditions for periodic steady state.

2.4. Problem formulation

In the following treatment we collect the periodic variables p , T , V and T_m in the vector y_p of length N_p . The part of y_p corresponding to p we denote y_p^s and the part of y_p corresponding to T , V and T_m we denote y_p^{ns} . We denote the integration variables E_w in the integral conditions for the wall temperatures as the vector y_i of length N_i , and the constant wall temperatures as the vector c_i . Finally, we denote the integration variable for the mean pressure needed for scaling of solutions as y_s . We want to find the initial values $y_{p,0}$ and the parameters c_i so that we simultaneously satisfy the cyclic boundary conditions for y_p , the integral conditions for the final values of y_i , and the condition for the final value of y_s .

3. Method

3.1. Formulation of shooting methods

To define a single shooting method we formulate the conditions for periodic steady state for $y_{p,0}$ and c_i as residual equations. We denote the starting time of the cycle t_0 and the cycle period Δt . By the notation $y_p(t_0 + \Delta t; y_{p,0}, c_i)$ we mean the values of y_p at time $t_0 + \Delta t$ given initial values $y_{p,0}$ and parameters c_i . We label the wanted final values of y_i as $ICTV$ (Integral Condition Target Values) and the wanted final value of y_s as $SCTV$ (Scaling Condition Target Value). Using this notation and introducing a scaling factor, λ , as an additional variable we can write the system of residual equations as

$$\begin{aligned} res_p(y_{p,0}, c_i) &= \begin{cases} y_{p,0}^s - y_p^s(t_0 + \Delta t; y_{p,0}, c_i) + (1 - \lambda) \cdot y_{p,0}^s \\ y_{p,0}^{ns} - y_p^{ns}(t_0 + \Delta t; y_{p,0}, c_i) \end{cases} \\ res_i(y_{p,0}, c_i) &= ICTV - y_i(t_0 + \Delta t; y_{p,0}, c_i) \\ res_s(y_{p,0}, c_i) &= SCTV - \lambda \cdot y_s(t_0 + \Delta t; y_{p,0}, c_i) \end{aligned} \quad (1)$$

The last term in the residual equations for $y_{p,0}^s$ is a penalty factor that forces λ towards a value of one. We choose to solve this system of $N_p + N_i + 1$ equations for the $N_p + N_i$ values in $y_{p,0}$ and c_i using a modified Newton–Raphson method. We denote the Jacobian matrix for the equation system (1) as $\underline{J_s}$ and the residuals of (1) as $\underline{r_s}$. Due to the complexity of the models that must be handled by the shooting method $\underline{J_s}$ cannot be calculated analytically and must be calculated by numerical differencing. This means that we must perform, at least, one integration of the model in order to evaluate a column in $\underline{J_s}$. In the iterations of the modified Newton–Raphson method we set $\lambda = 1$ and then solve a linear system and correct our guess for the solution as

$$\underline{J_s} \cdot \begin{bmatrix} \frac{\delta y_{p,0}}{\delta c_i} \\ \frac{\delta \lambda}{\delta c_i} \end{bmatrix} = -\underline{r_s}, \quad \begin{bmatrix} y_{p,0}^s \\ y_{p,0}^{ns} \\ c_i \end{bmatrix} = \begin{bmatrix} y_{p,0}^s \\ y_{p,0}^{ns} \\ c_i \end{bmatrix} + \begin{bmatrix} \frac{\delta y_{p,0}^s}{\delta y_{p,0}^{ns}} \\ \frac{\delta y_{p,0}^{ns}}{\delta y_{p,0}^{ns}} \\ \frac{\delta c_i}{\delta c_i} \end{bmatrix} \quad (2)$$

The modification of the Newton–Raphson method is, hence, that we keep setting $\lambda = 1$ and thereby ignore $\Delta\lambda$.

In the single shooting method defined above we shoot across an entire cycle of the engine. We can extend the single shooting method to a multiple shooting method by slicing the cycle into multiple subintervals by introducing cuts at N_{ip} internal points. Additional equations for the continuity of the solution are then needed. If we choose to number the resulting $N_{ip} + 1$ intervals from 0 to N_{ip} and to label the variables belonging to interval j with subscript (j) we can write the complete system of residual equations for the multiple shooting method as

$$\begin{aligned} \underline{res_p}(y_{(N_{ip}),p,0}, y_{(0),p,0}, c_i) &= \begin{cases} y_{(0),p,0}^s - y_{(N_{ip}),p}^s(t_{N_{ip}} + \Delta t_{N_{ip}}; y_{(N_{ip}),p,0}, c_i) + (1 - \lambda) \cdot y_{(0),p,0}^s \\ y_{(0),p,0}^{ns} - y_{(N_{ip}),p}^{ns}(t_{N_{ip}} + \Delta t_{N_{ip}}; y_{(N_{ip}),p,0}, c_i) \end{cases} \\ \underline{res_{(j)}}(y_{(j-1),0}, y_{(j),0}, c_i) &= \begin{cases} y_{(j),p,0}^s - y_{(j-1),p}^s(t_{j-1} + \Delta t_{j-1}; y_{(j-1),p,0}, c_i) + (1 - \lambda) \cdot y_{(j-1),p,0}^s \\ y_{(j),p,0}^{ns} - y_{(j-1),p}^{ns}(t_{j-1} + \Delta t_{j-1}; y_{(j-1),p,0}, c_i) \\ y_{(j),i,0} - y_{(j-1),i}(t_{j-1} + \Delta t_{j-1}; y_{(j-1),0}, c_i) \\ y_{(j),s,0} - y_{(j-1),s}(t_{j-1} + \Delta t_{j-1}; y_{(j-1),0}, c_i) \end{cases} \quad j = 1..N_{ip} \quad (3) \\ \underline{res_{ic}}(y_{(N_{ip}),p,0}, y_{(N_{ip}),i,0}, c_i) &= \underline{ICTV} - y_{(N_{ip}),i}(t_{N_{ip}} + \Delta t_{N_{ip}}; y_{(N_{ip}),p,0}, y_{(N_{ip}),i,0}, c_i) \\ \underline{res_s}(y_{(N_{ip}),p,0}, y_{(N_{ip}),s,0}, c_i) &= \underline{SCTV} - \lambda \cdot y_{(N_{ip}),s}(t_{N_{ip}} + \Delta t_{N_{ip}}; y_{(N_{ip}),p,0}, y_{(N_{ip}),s,0}, c_i) \end{aligned}$$

We denote the Jacobian matrix for the equation system (3) as $\underline{J_{ms}}$ and the residuals of (3) as $\underline{r_{ms}}$. In the multiple shooting method we also choose to solve our residual equation system of $N_{ip} \cdot (N_p + N_i + 1)$ equations for the $N_{ip} \cdot (N_p + N_i + 1) - 1$ values in $y_{(0),p,0}$, c_i , and $y_{(j),0}$, $j = 1..N_{ip}$ using a modified Newton–Raphson method. In the modified Newton–Raphson iterations of the multiple shooting method we set $\lambda = 1$ and then solve a linear system and correct our guess for the solution as

$$\begin{aligned} \underline{J_{ms}} \cdot \begin{bmatrix} \frac{\delta y_{(0),p,0}}{\delta y_{(j)},} \\ \frac{\delta c_i}{\delta \lambda} \end{bmatrix} &= -\underline{r_{ms}}, \quad j = 1..N_{ip} \\ \begin{bmatrix} y_{(0),p,0}^s \\ y_{(0),p,0}^{ns} \\ y_{(j),0} \\ c_i \end{bmatrix} &= \begin{bmatrix} y_{(0),p,0}^s \\ y_{(0),p,0}^{ns} \\ y_{(j),0} \\ c_i \end{bmatrix} + \begin{bmatrix} \frac{\delta y_{(0),p,0}^s}{\delta y_{(0),p,0}^{ns}} \\ \frac{\delta y_{(0),p,0}^{ns}}{\delta y_{(0),p,0}^{ns}} \\ \frac{\delta y_{(j),0}}{\delta c_i} \\ \frac{\delta c_i}{\delta c_i} \end{bmatrix} \quad j = 1..N_{ip} \quad (4) \end{aligned}$$

When the equations are ordered as in (3) then the sparse structure of $\underline{J_{ms}}$ takes the form (5) where $\underline{I}$ is the identity matrix and the X symbols mark other non-zero entries:

$$J_{ms} = \begin{bmatrix} \underline{J(0),p,0} & \underline{J(1),0} & \underline{J(2),0} & \underline{J(3),0} & \cdots & \underline{J(N_p)} & \underline{c_i} & \lambda \\ \underline{I} & & & & & \underline{X} & \underline{X} & \underline{res_p} \\ \underline{X} & \underline{I} & & & & \underline{X} & \underline{X} & \underline{res(1)} \\ & \underline{X} & \underline{I} & & & \underline{X} & \underline{X} & \underline{res(2)} \\ & & \underline{X} & \underline{I} & & \underline{X} & \underline{X} & \underline{res(3)} \\ & & & \ddots & \ddots & \vdots & \vdots & \vdots \\ & & & & \underline{X} & \underline{I} & \underline{X} & \underline{res(N_p)} \\ & & & & & \underline{X} & \underline{X} & \underline{res_i} \\ & & & & & \underline{X} & \underline{X} & \underline{res_s} \end{bmatrix} \quad (5)$$

3.2. Reducing the dimension of the boundary value problem with single shooting

When we perform single shooting we can sometimes use insight into the time scales of the physical phenomena being modelled to reduce the dimension of the BVP handled by the shooting method. In the Stirling engine model introduced above changes in $\underline{p}$ and $\underline{V}$ will propagate with the speed of sound. Pressure waves will cause small fluctuations in $\underline{T}$ but the major changes in $\underline{T}$ oscillate with a period equal to the length of the cycle propagate with the bulk flow velocities induced by the movements of the pistons; the bulk velocities are at least an order of magnitude smaller than the speed of sound in the working gas. Changes in $\underline{T_m}$ oscillate slowly with a period of equal to the length of the cycle. It is primarily the evolution of $\underline{T_m}$ and $\underline{T}$ that cause the slow evolution of solutions to the model.

If we choose to exclude $\underline{p}$ and $\underline{V}$ from the single shooting method then we need to perform direct iteration on the corresponding elements of $\underline{y_p}$ when we iterate the shooting method. We remove the equations corresponding to the excluded variables and $\underline{y_s}$ from (1), and we perform direct iteration by changing the corrections for $\underline{y_p}$ and $\underline{c_i}$ in the modified Newton–Raphson iterations in (2) to:

$$\begin{bmatrix} \underline{J_{p,0}^s} \\ \underline{J_{p,0}^{ns}} \\ \underline{J_{p,0}^{ns}} \\ \underline{c_i} \end{bmatrix} = \begin{bmatrix} \underline{J_{p,0}^s} \\ \underline{J_{p,0}^{ns}} \\ \underline{J_{p,0}^{ns}} \\ \underline{c_i} \end{bmatrix} + \left\{ \begin{array}{l} \left[\frac{SCTV}{\underline{y_s} \left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i} \right)} \cdot \underline{y_p^s} \left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i} \right) - \underline{J_{p,0}^s} \right] \\ \left(\underline{y_p^{ns}} \left(t_0 + \Delta t; \underline{y_{p,0}}, \underline{c_i} \right) - \underline{J_{p,0}^{ns}} \right) \\ \frac{\delta \underline{J_{p,0}^{ns}}}{\delta \underline{c_i}} \end{array} \right\} \begin{array}{l} \text{excluded from shooting} \\ \\ \text{included in shooting} \end{array} \quad (6)$$

By reducing the number of variables included in the shooting method the effort needed to compute $\underline{J_s}$ is reduced. For the Stirling engine model this approach has the additional benefit that it can stabilize the solution process because including $\underline{p}$ in the shooting process can cause problems. This is because a change in a single element in $\underline{p_0}$ will distribute rapidly between all elements of $\underline{p}$ and will change all the elements of $\underline{res_p}$ by almost equal amounts. This can make it more difficult to achieve convergence if starting from a very bad pressure distribution.

Unfortunately, there appears to be no obvious way to extend this method for reducing the dimension of the BVP to the multiple shooting method. The problem is, that there appears to be no efficient way to perform direct iteration on the excluded variables simultaneously in multiple intervals.

It should be noted that we have observed that excluding $\underline{p}$ and $\underline{V}$ appears to work well for models of Stirling machines in general, but that it does not work well for models of Stirling type pulse tube coolers (Stirling type PTCs). The geometries of Stirling type PTCs are such that equalisation of a pressure difference between the ends of the gas domain takes a larger fraction of a cycle than a corresponding pressure equalisation in a Stirling engine. Therefore the evolution of $\underline{p}$ is more important to the periodic steady state solution in a Stirling type PTC and therefore only $\underline{V}$ should be excluded when finding periodic steady state solutions to models of Stirling type PTCs.

3.3. Comparison of work for shooting methods

The above shooting methods all require one simulation of the entire cycle of length Δt in order to update the residuals. But the total amounts of running time of the engine that must be simulated in order to compute a Jacobian matrix differ for the methods.

If we calculate the elements of the Jacobian using one sided differences and if we already know the current residuals, then we need to simulate the entire cycle $N_p + N_i + 1$ times in order to compute $\underline{J}_s$ for the single shooting method. If reduce the dimension of the boundary value problem by omitting $\underline{p}$ and $\underline{V}$ we reduce the required number of simulations by the number of pressures and velocities plus one. Determining the number of simulations required to calculate $\underline{J}_{ms}$ for the multiple shooting method is slightly more complex. Calculating the non-zero elements corresponding to the X symbols in (5) requires $(N_{ip} + 1) \cdot (N_p + N_i + 1) + N_{ip} \cdot (N_i + 1)$ simulations of subintervals. The term $(N_{ip} + 1) \cdot (N_p + N_i + 1)$ is the effort needed to simulate the entire cycle $N_p + N_i + 1$ times, corresponding to the effort needed to compute $\underline{J}_s$. For some models the overhead corresponding to the term $N_{ip} \cdot (N_i + 1)$ can be fully or partially avoided. If the values of y_i and y_s do not affect y_p during the simulation then $N_{ip} \cdot (N_i + 1)$ perturbations of y_i and y_s at the internal points in the cycle can be omitted during updates of $\underline{J}_{ms}$. Any changes in the initial values of y_i and y_s in a subinterval will transfer directly to the final values of y_i and y_s in the same subinterval when the subinterval is simulated. The corresponding elements of unit magnitude can be inserted directly in $\underline{J}_{ms}$.

Even when the total engine running time that must be simulated is the same for Jacobian updates in the single and multiple shooting methods $\underline{J}_{ms}$ can still be more costly to evaluate than $\underline{J}_s$ because of the increased number of simulation start ups. A minor overhead is due to restarting the IVP method and its step size control algorithm but a much more significant overhead can be caused by the fast transients described in the introduction with acoustic waves in the beginning of the simulations. For the Stirling engine model used here the fast transients caused by perturbations of elements of $\underline{p}$ and by poor initial guesses are significant.

3.4. Reducing computational work

In order to minimise the overhead caused by the fast transients a single simulation is run to smooth the initial pressure distributions before either $\underline{J}_s$ or $\underline{J}_{ms}$ are computed. This helps the most when $\underline{J}_s$ and, especially, $\underline{J}_{ms}$ are computed from a poor initial guess so that every subinterval begins with a large transient. The smoothing effect is obtained by simulating all subintervals in the cycle and then cyclically shifting the final values to be new initial values in the following way:

$$\begin{aligned} \underline{y}_{(N_{ip}),p}(t_{N_{ip}} + \Delta t_{N_{ip}}) &\text{ transfers to } \underline{y}_{(0),p,0} \\ \underline{y}_{(j-1),p}(t_{j-1} + \Delta t_{j-1}) &\text{ transfers to } \underline{y}_{(j),0}, \quad j = 1..N_{ip} \end{aligned} \quad (7)$$

Also, in order to begin the shooting from as close to the correct pressure distribution as possible two smoothing simulation runs are performed where $\underline{y}_0$ is updated as shown in (8), before beginning the actual shooting:

$$\left. \begin{aligned} \frac{SCTV}{y_s(t_{N_{ip}} + \Delta t_{N_{ip}})} \cdot \underline{y}_{(N_{ip}),p}^s(t_{N_{ip}} + \Delta t_{N_{ip}}) &\text{ transfers to } \underline{y}_{(0),p,0}^s \\ \underline{y}_{(N_{ip}),p}^{ns}(t_{N_{ip}} + \Delta t_{N_{ip}}) &\text{ transfers to } \underline{y}_{(0),p,0}^{ns} \\ \frac{SCTV}{y_s(t_{N_{ip}} + \Delta t_{N_{ip}})} \cdot \underline{y}_{(j-1),p}^s(t_{j-1} + \Delta t_{j-1}) &\text{ transfers to } \underline{y}_{(j),p,0}^s \\ \underline{y}_{(j-1),p}^{ns}(t_{j-1} + \Delta t_{j-1}) &\text{ transfers to } \underline{y}_{(j),p,0}^{ns} \\ \underline{y}_{(j-1),i}(t_{j-1} + \Delta t_{j-1}) &\text{ transfers to } \underline{y}_{(j),i,0} \\ \underline{y}_{(j-1),s}(t_{j-1} + \Delta t_{j-1}) &\text{ transfers to } \underline{y}_{(j),s,0} \end{aligned} \right\} j = 1..N_{ip} \quad (8)$$

When iterating the shooting method for Stirling machine models it is usually possible to recycle the Jacobian matrix for multiple iterations and even between multiple solutions to the same model if convergence remains

satisfactory. A robust criterion for the recycling is needed and here it has been chosen to base the criterion on the L^2 -norms of four groups of residuals: res_p , res_i , res_s , and $res_{(j)}$, $j = 1..N_{ip}$. When none of the four L^2 -norms have decreased by a factor of at least $JacRecycleCrit$ in either of the last two iterations then the Jacobian matrix is updated. This requirement is not very strict and it leaves the possibility that the solution can get stuck or diverge at some point during the iterations with the individual residuals norms alternately increasing and decreasing in such a way that the Jacobian matrix is not updated. To rule out this possibility the maximum residual is continuously monitored and if it has not decreased in 10 iterations then the Jacobian is updated.

3.5. Parallelisation of shooting methods

When the shooting methods discussed above were applied to the Stirling engine model nearly all the CPU-time needed by the methods was spent in the IVP method used for integrating the model. Hence the efforts to parallelise the shooting methods were concentrated on the parts of the methods that require simulations to be performed: Performing smoothing simulation runs, calculating residuals, and updating Jacobians. Performing smoothing runs as shown in (7) and (8) and updating the residuals all require the simulation of one cycle. Hence all simulation capabilities needed by the shooting methods was encapsulated in two subroutines: One that computes $y_{(j)}(t_j + \Delta t_j)$, $j = 0..N_{ip}$ and one that computes $\underline{J}_s$ or $\underline{J}_{ms}$. Parallel and sequential versions have been made of these two subroutines.

The two subroutines that perform simulation tasks for the shooting methods can be parallelised with different levels of granularity. At the fine grained level one could parallelise the IVP method called from the shooting method. Based on a study by Bendtsen [19] and experiments with parallelisation of the IVP method this was reckoned to result in too fine a granularity to scale well on distributed memory architectures for the relevant moderate size of IVPs and hence was not further explored. The two subroutines were instead parallelised with a much coarser granularity by making them distribute simulations of subintervals between multiple processes. For both of the subroutines the number of simulations that can be distributed increases when N_{ip} is increased and the achievable speedup from parallelisation should thus also increase with N_{ip} . The parallelisation of the two subroutines was done using the pool of tasks paradigm [20] where a master process tries to keep a number of worker processes busy by distributing independent tasks to idle processes until all tasks in the pool of tasks is complete.

The shooting method, the IVP method, and the Stirling engine model are implemented in ISO standard Fortran 95. The software has been tested to be compatible with Fortran 95 compilers from Compaq, Sun, Lahey/Fujitsu, Intel, HP and IBM and thus runs on a large number of platforms. The parallelisation has been done using the Message Passing Interface library (MPI) in order to preserve portability to distributed memory platforms.

3.6. Tests case

In order to compare the shooting methods on a relevant basis we used an actual production run as test job. The test job was a batch of 28 simulations for the Stirling engine model and corresponds to a set of experimental data where the periodic steady state performance of the SM5 Stirling engine has been measured at 28 distinct operating conditions. The experimental data is distributed over a wide range of operating conditions with helium or nitrogen as working gas. The test job is started without a Jacobian from a common reference solution not included in the test job. The test job was considered difficult with regards to recycling of Jacobians because the solutions span a wide range of temperatures, a wide range of ratios between gas and matrix heat capacities, and because flow friction is much larger when nitrogen is used as working gas instead of helium. The number of control volumes in the Stirling engine model was chosen so that $N_p = 319$ and $N_i = 78$ giving a total of 398 variables in the IVP.

In the test job the measure for the accuracy of the solutions was based on the imbalance in the overall energy balance for the engine. The overall energy balance is calculated from the heat absorbed by the heater tubes, the electrical power output from the built in generator of the engine, and the heat rejected to the cooling water by the engine and the generator. The overall energy balance gives a very good indication of how close a

solution is to periodic steady state because it reveals if the regenerator matrix is absorbing or giving of small amounts of energy from cycle to cycle, even if the large heat capacity of the matrix hides this in the elements of res_p corresponding to T_m . For each solution shooting was terminated when the absolute value of the imbalance in the overall energy balance was less than 0.1 W.

3.7. Benchmarking procedure

For the multiple shooting method it was chosen to place, respectively, 1, 3, 7, and 15 internal points in the cycle. To achieve good load balancing the internal points were placed so that the resulting subintervals of the cycle required approximately the same computational effort to simulate. The division was made by running one simulation from an accurate reference solution and storing the position in the cycle and the wall time at every step taken by the IVP method. The positions of the internal points were then determined by interpolation in the stored data.

During the Jacobian updates it was exploited that y_i and y_s do not affect other values of y during simulations so that some perturbations could be skipped. However, it was chosen to skip the perturbations of y_i and y_s only at the first $N_{ip} - 1$ internal points, so that the effects of structural heat conduction computations that change the values of y_i after simulation of subinterval number N_{ip} could be included in the Jacobian matrix.

To compare the performance of the methods the test job was completed in the following tests:

1. Sequential test without acceleration of periodic variables towards steady state.
2. Sequential test using each of the shooting methods in turn.
3. Parallel test using each of the shooting methods in turn.

During the tests the total running times, profiles, and the numbers of evaluations were recorded. The solutions for each of the methods have been tested to be bitwise identical independently of the number of worker processes and the solution processes are thus directly comparable.

The first test with no acceleration of the periodic variables towards periodic steady state was performed to have a base case against which to compare the shooting methods. The test was carried out by excluding all elements of y_p from the single shooting method and then iterating as described in (6) so that the shooting method was used only to solve the integral conditions for T_w . This approach can be considered a hybrid between integration to convergence and the shooting method but it is just denoted as integration to convergence from hereon. In this test the strategy for updating Jacobians was changed so that the Jacobian was updated only if the imbalance in the overall energy balance had not decreased for two iterations in a row.

The measurements of the sequential performance of the shooting methods were carried out in order to compare the computational costs of the methods without incurring any communications overhead. The average time required to simulate a revolution and to perform a Jacobian update were then calculated for each method in the first two tests.

Table 1
Parallel characteristics of methods included in the numerical tests

Name of method	Variables excluded from shooting	N_{ip}	Tasks in simulation of a cycle	Tasks in a Jacobian update
Integration to convergence	y_p (319 vars.)	0	1	78
Single shooting, excl. P , V	p, V (191 vars.)	0	1	206
Single shooting	None	0	1	398
Multiple shooting, 2 int.	None	1	2	875
Multiple shooting, 4 int.	None	3	4	1671
Multiple shooting, 8 int.	None	7	8	3263
Multiple shooting, 16 int.	None	15	16	6447

The parallel test was performed to measure the scalabilities of the shooting methods. The scalabilities of the shooting methods have been measured by running them on 3, 5, 9, 17, and 33 processors corresponding to 2, 4, 8, 16, and 32 worker processes being available to perform tasks for the master process. We assess the scalability of a shooting method by the absolute speed up, calculated as the sequential running time for the method divided by the parallel running time of the method. We compare the parallel methods using the relative speed up, defined as the running time of the fastest sequential method divided by the running time in the present experiment. The numbers of tasks in a simulation of a cycle and in a Jacobian update are summarised for the tested methods in Table 1.

The semi-implicit generalised Runge–Kutta method GERK of order three by Thomsen [21] was used as the IVP method needed to integrate the Stirling engine model. A relative tolerance of 10^{-7} was used for the error pr. step taken by the GERK method, and a relative residual tolerance of 10^{-11} was used by the Newton–Raphson solver used at the stages of the GERK method. The value of *JacRecycleCrit* was set to 0.85 for all methods, except for integration to convergence where the criterion was changed as described above.

All tests were run on a Sunfire15000 server with 42 UltraSparc III processors operating at 1.05 GHz and running the Solaris 9 operating system. The Sunfire15000 used for the testing is a shared memory machine and the communications overhead from the MPI subroutine calls should thus be minimal. The timing results contain a small amount of noise because the tests were run in a multi-user environment.

4. Numerical results

Fig. 3 shows an example of how the residual norms and the imbalance in the overall energy balance of the engine decreased when a solution to the Stirling engine model was found by integration to convergence. This particular solution had helium at 6.5 MPa mean pressure as the working gas of the engine.

The results of the sequential tests of integration to convergence and of the shooting methods are shown in Table 2.

The absolute speed ups calculated from the results of the parallel tests are shown in Table 3.

The speed ups relative to the sequential running time of the single shooting method with $\underline{p}$ and $\underline{V}$ excluded from the shooting are shown in Fig. 4.

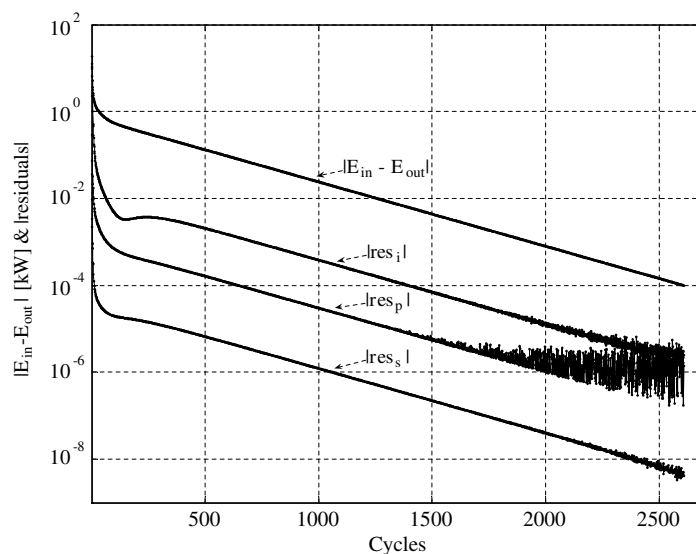


Fig. 3. Example of the decrease in the residual norms and the imbalance in the energy balance of the engine when performing integration to convergence.

Table 2

Results from sequential tests of integration to convergence and the shooting methods

Method	Running time		Profile					Evaluations				Avrg. cost of evals.	
	Time [h]	Index [-]	Resid. [%]	Sm. r. [%]	Jac. [%]	LA [%]	Other [%]	Iter. [-]	Resid. [-]	Sm. r. [-]	Jac. [-]	Sim. cycle [s]	Jac. calc. [h]
Integration to convergence	472.4	100	99.2	0.1	0.2	0.0	0.5	69271	69301	58	2	24.3	0.50
Single shooting, excl. P , V	4.4	0.9	34.1	8.8	61.1	0.0	0.1	196	226	58	2	24.1	1.36
Single shooting	7.0	1.5	22.2	5.6	84.3	0.0	0.0	201	231	58	2	24.4	2.97
Multiple shooting, 2 int.	12.8	2.7	14.6	3.3	88.9	0.0	0.1	215	265	59	3	25.4	3.79
Multiple shooting, 4 int.	16.6	3.5	12.8	2.8	84.3	0.1	0.1	227	284	59	3	27.1	4.66
Multiple shooting, 8 int.	32.3	6.8	10.4	1.8	87.6	0.2	0.1	316	352	60	4	34.3	7.08
Multiple shooting, 16 int.	84.0	17.8	6.4	1.0	92.3	0.2	0.0	307	345	63	7	55.0	11.08

The profiles of the methods show the percentages of the running time spent calculating residuals, making smoothing simulation runs, computing Jacobians, performing linear algebra with the Jacobians, and the time spent on other tasks. The columns under the heading Evaluations show the number of iterations, residual evaluations, smoothing runs, and Jacobians needed by the methods in order to complete the test job. The rightmost columns shows the average time needed per evaluation when simulating the entire cycle once and when updating the Jacobian matrix.

Table 3

Absolute speed up for simulating cycles, for computing Jacobians, and for the complete test job for each of the shooting methods

Method	Speed up: Simulation of cycle					Speed up: Comput. Jacobians					Overall speed up				
	Number of worker processes					Number of worker processes					Number of worker processes				
	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32
Single shooting, excl. P , V	1.0	1.0	1.0	1.0	0.9	2.0	4.0	7.9	15.7	29.1	1.4	1.8	2.0	2.2	2.2
Single shooting	1.0	1.0	1.0	1.0	1.0	2.0	4.0	8.0	15.4	30.3	1.6	2.3	2.9	3.3	3.7
Multiple shooting, 2 int.	2.0	2.0	1.9	2.0	2.0	2.0	4.0	7.9	15.9	31.3	2.0	3.4	5.2	7.2	8.9
Multiple shooting, 4 int.	1.9	3.5	3.5	3.5	3.5	2.0	4.0	7.9	16.0	31.5	2.0	3.9	6.6	10.1	13.7
Multiple shooting, 8 int.	1.9	3.6	6.0	6.1	6.1	2.0	4.0	7.9	15.9	31.6	2.0	3.9	7.5	13.0	20.1
Multiple shooting, 16 int.	2.0	3.6	5.8	7.3	7.2	2.0	4.0	8.0	15.9	31.1	2.0	4.0	7.7	14.1	23.0

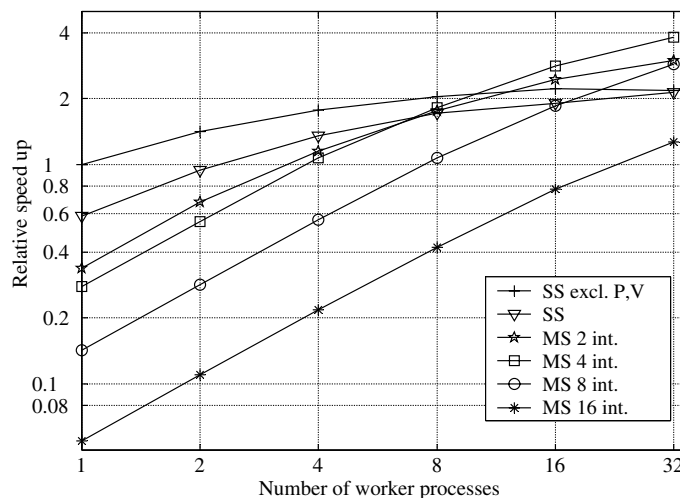


Fig. 4. Speed ups relative to sequential single shooting with reduced dimension of BVP achieved with shooting methods.

5. Discussion

5.1. Integration to convergence

Fig. 3 shows the decreases in residuals and the imbalance of the overall energy balance during integration to convergence for a solution with helium at 6.5 MPa mean pressure as the working gas. Simulation of approximately 2600 cycles was needed to converge. The SM5 Stirling engine operates at 1025 RPM so this corresponds to approximately 2.5 min running time for the engine. After the initial 500 iterations the residual norms and the imbalance in the overall energy balance of the engine appeared to decrease linearly in the semi-logarithmic plot in Fig. 3 indicating that the solution approached periodic steady state exponentially. After approximately 1500 iterations random noise began to show in the residual norms indicating that the changes from the evolution of the solution during each cycle approached the level of noise due to the chosen accuracy of the IVP method. The noise was most significant for res_p that contained the largest number of variables and included the fast components p and V . The imbalance in the overall energy balance for the engine, though, still decreased monotonously. If the residual norms had been used to determine if J_s should be recycled, the noise would have caused J_s to be updated more frequently than what was needed here to obtain monotonous decrease in the imbalance in the overall energy balance. Hence the choice of basing the criterion for recycling Jacobians on the imbalance in the overall energy balance in this test was appropriate.

The number of iterations required to reach the desired accuracy was strongly influenced by the ratio of the heat capacity of the gas to the heat capacity of the regenerator matrix. Integration to convergence with helium at 5 MPa and 8 MPa mean pressures required approximately 3200 and 2100 iterations, respectively. Integration to convergence with nitrogen at 6.5 MPa and 8 MPa mean pressure required approximately 1400 and 1200 iterations to converge.

5.2. Comparison of results for sequential methods

Table 2 shows that the computational effort required to complete the test job was significantly smaller for the shooting methods than for integration to convergence. The table also shows that the fastest sequential method was the single shooting method where the dimension of the BVP had been reduced by excluding p and V from the shooting. Table 2 shows that the number of needed iterations and Jacobian evaluations increased with N_{ip} and that the evaluations became significantly more costly as N_{ip} was increased. Jacobian updates typically occurred when changing the working gas or at large differences in mean pressures between solutions.

The profile in Table 2 shows that more than 99% of the CPU-time was spent calculating residuals when integration to convergence was performed. For the shooting methods the time spent computing Jacobians was more significant. It took approximately 61% of the total running time for the single shooting method with reduced dimension of the BVP and it increased with N_{ip} to approximately 92% for the multiple shooting method with $N_{ip} = 15$. The time spent performing linear algebra on J_s and J_{ms} in the shooting methods was not significant.

The reasons for the negative influences of increasing N_{ip} were related to the Stirling engine model that models a fully compressible flow problem where acoustic phenomena are present in the solutions. Acoustic waves were induced when simulations were started from bumpy initial pressure distributions. As long as acoustic waves were present the IVP method applied a very fine temporal discretisation. The number of such transients increases with N_{ip} and this was the main reason for the increase in the computational effort required to perform simulations.

When the subintervals of the cycle became short enough the acoustic waves did not have time to die out completely during the simulations of the subintervals. The waves polluted the Jacobians because they were present only during the numerical differencing performed to compute the elements of the Jacobians and not in the periodic steady state solutions. This was believed to be the main reason for the poorer convergence observed as N_{ip} was increased. As an experiment N_{ip} was increased to 31 and with this discretisation convergence was not achieved. The shooting methods presented here have previously been applied to an earlier version of the Stirling engine model [5], which did not include the effects of gas inertia in the momentum equation,

and where pressure waves were dissipated much faster. When the simpler Stirling engine model was configured to model the SM5 Stirling engine multiple shooting was successfully applied with as many as 500 subintervals in the cycle. A similar, but much smaller, increase in computational effort per simulation task with increasing N_{ip} was observed in these experiments but no convergence problems related to short subintervals were observed for the simpler model. This supports that the fast transients with acoustic waves are the reason for the negative influence of increasing N_{ip} .

5.3. Comparison of parallel methods

Table 3 shows that the parallel computations of $\underline{J_s}$ and $\underline{J_{ms}}$ achieved near linear speed up. This shows that the communications overhead was small and that good load balancing was achieved. The simulations of the entire cycle achieved lesser speed ups. The best possible speed up for a simulation of the cycle is $(N_{ip} + 1, N_{\text{worker processes}})$ if perfect load balancing is achieved and significantly smaller speed ups were achieved for $N_{ip} > 1$ on more than two processors. The limited speed ups were mainly due to imperfect load balancing. The imperfect load balancing was caused by the fast transients in the beginning of the subintervals requiring different amounts of computational effort to simulate. Table 3 also shows that the absolute speed ups achieved by the methods improved with N_{ip} . One reason for this is that the speed ups achieved for the simulations of the entire cycle were higher for larger N_{ip} and another reason was that computations of Jacobians took up a larger fraction of the total running time as N_{ip} was increased.

Fig. 4 shows that the multiple shooting method with $N_{ip} = 3$ on 32 processors achieved the largest relative speed up of all the tested methods and that this relative speed up was only 3.8 for the test job.

6. Conclusion

Single and multiple shooting methods for finding steady state periodic solutions to one dimensional models of reciprocating machines, such as Stirling machines and pulse tube coolers, have been developed and parallelisation of the algorithms has been explored. The methods have been compared using a test job consisting of a batch of simulations with a Stirling engine model.

The tested methods were found to be reliable for finding periodic steady state solutions for the test job. The single shooting method with reduced dimension of the boundary value problem was found to be the fastest of the tested sequential methods and it also exhibited the best convergence properties. It required less than 1% of the computational efforts needed to complete the test job without accelerating the convergence of the periodic variables towards periodic steady state.

It was observed that the computational efforts needed by the shooting methods increased with the number of internal points in the cycle, i.e. with the number of subintervals in the cycle, for the Stirling engine model used for the tests. At the same time the scalabilities of the shooting methods on parallel computers improved with the number of internal points in the cycle. The largest absolute speed up of 23 on 33 processors was observed for the multiple shooting method with 15 internal points in the cycle, i.e. with the cycle divided into 16 subintervals. Due to the overhead in the shooting methods caused by increasing N_{ip} the best achieved speed up relative to the fastest sequential method was only 3.8 and was achieved for multiple shooting with 3 internal points in the cycle using 33 processors.

The large overheads for the multiple shooting methods reported in this paper were due to the characteristics of the model used for the testing, more specifically to fast transients that occurred while acoustic waves were dissipated in the beginning of the IVPs solved internally in the shooting method. The multiple shooting methods appear more attractive when applied to models where transients due to permutations of initial values are not as computationally expensive and are short lived relative to the lengths of the subintervals in the cycle.

Acknowledgements

The hardware used for the numerical experiments was made available by the Sun High Performance Computing (HPC) Systems at the Technical University of Denmark. The installation is part of the DCSC Center

for Scientific Computing – DTU, which operates the computer installations on behalf of the Danish Center for Scientific Computing.

The initial development of the simulation software was sponsored by the Danish energy company DONG.

References

- [1] V.V. Toropov, H. Carlsen, Optimization of Stirling engine performance based on multipoint approximation technique, in: B.J. Gilmore, et al. (Eds.), *Advances in Design Automation 1994, Robust Design Applications, Decomposition and Design Optimization, Optimization Tools and Applications*, 20th Design Automation Conference, Minneapolis, vol. 2, September 11–14, 1994, pp. 531–536.
- [2] I. Urieli, A current review of Stirling cycle machine analysis methods, in: *Proceedings of 18th IECEC*, vol. 2, American Institute of Chemical Engineers, Orlando, 1983, pp. 702–707.
- [3] K.B. Wilson, D.R. Gedeon, Status of Pulse Tube Cryocooler Development at Sunpower, Inc., 2004, see also: <http://www.sunpower.com/lib/sitefiles/pdf/publications/Doc0089.pdf>.
- [4] H. Carlsen, J. Bovin, Test of 9 kW Stirling engine using biogas as fuel, in: *Tenth International Stirling Engine Conference*, VDI, Osnabrück, September, 2001.
- [5] S.K. Andersen, H. Carlsen, P.G. Thomsen, Preliminary results from simulations of temperature oscillations in Stirling engine regenerator matrices, *Energy* 31 (2006) 1371–1383.
- [6] S.K. Andersen, H. Carlsen, P.G. Thomsen, Control volume based modelling in one space dimension of oscillating, compressible flow in reciprocating machines, *Simulation Modelling Practice and Theory* 14 (2006) 1073–1086.
- [7] S. Skelboe, Extrapolation methods for computation of the periodic steady-state response of non-linear circuits, in: *Proceedings of IEEE International Symposium on Circuits and Systems*, 1977, pp. 64–67.
- [8] S. Skelboe, Computation of the periodic steady-state response of non-linear networks by extrapolation methods, *IEEE Transactions on Circuits and Systems* CAS-27 (3) (1980) 161–185.
- [9] H.D. Kühl, Veralgemeinerte thermodynamische Beschreibung regenerativer Gaskreisprozesse, Doctoral thesis, Fortschritt Berichte VDI Reihe 19 Nr. 42, VDI-Verlag GmbH, Düsseldorf, 1990.
- [10] S.K. Andersen, H. Carlsen, P.G. Thomsen, Numerical study on the appendix gap losses in a Stirling engine, in: *Proceedings of the 12th International Stirling Engine Conference and Technology Exhibition*, Durham UK, 7–9 September, 2005, pp. 336–347, ISBN: 0-9535-5582-8.
- [11] J.S. Rauch, Harmonic analysis of Stirling engine thermodynamics, in: *Proceedings of 15th IECEC*, vol. 2, American Institute of Aeronautics and Astronautics, Seattle, 1980, pp. 1696–1700.
- [12] S.C. Huang, HFAST – a harmonic analysis program for Stirling cycles, in: *Proceedings of 27th Intersociety Energy Conversion Engineering Conference*, vol. 5, Soc. Automotive Eng., 1992, pp. 547–552.
- [13] NASA Glenn Research Center, Thermo-electrical Systems Branch, Stirling Performance Codes, 2004, see also: <http://www.grc.nasa.gov/WWW/tmsb/stirling/doc/codes.html>.
- [14] D. Gedeon, A globally-implicit Stirling cycle simulation, in: *Proceedings of 21st IECEC*, vol. 1, American Chemical Society, San Diego, 1986, pp. 550–554.
- [15] L. Bauwens, Consistency, stability, convergence of Stirling engine models, *Proceedings of 25th IECEC*, vol. 5, American Institute of Chemical Engineers, 1990, pp. 352–358.
- [16] M.B. Commiso, Analysis of cyclic thermodynamic processes, Ph.D. Thesis, Department of Energy Engineering, Technical University of Denmark, Denmark, 1994.
- [17] T. Kato, H. Minato, M. Tanaka, Parallel analysis of a power electronic circuit, in: *Proceedings of the 3rd International Power Electronics and Motion Control Conference, IPEMC 2000*, vol. 2, 2000, pp. 654–659.
- [18] L. Siyun, H. Hofmann, Numerically efficient steady-state finite element analysis of magnetically saturated electromechanical devices using a shooting-Newton/GMRES approach, in: *Proceedings of IEEE International Electric Machines and Drives Conference, IEMDC 2001*, pp. 275–279.
- [19] C. Bendtsen, Parallel numerical algorithms for the solution of systems of ordinary differential equations, Ph.D. Thesis, Department of Informatics And Mathematical Modelling, Technical University of Denmark, Denmark, 1996.
- [20] B. Wilkinson, M. Allen, *Parallel Programming – Techniques and Applications Using Networked Workstations and Computers*, Prentice Hall, 1999.
- [21] P.G. Thomsen, A generalized Runge Kutta method of order three, Technical Report IMM-REP-2002-07, Department of Informatics And Mathematical Modelling, Technical University of Denmark, 2002.